Short Paper

Elman Network with Embedded Memory for System Identification

ADEM KALINLI AND SEREF SAGIROGLU*
Department of Industrial Electronics
Kayseri Vocational High School
Erciyes University
38039, Kayseri, Turkey
**Department of Computer Engineering*
Engineering and Architecture Faculty
Gazi University
06570, Ankara, Turkey

This paper presents a new recurrent neural network (RNN) structure called ENEM for dynamic system identification. ENEM structure is based on Elman network and NARX neural network. In order to show the performance of ENEM for system identification, the results were also compared to the results of Elman network, Jordan network and their modified models. The identification results of linear and nonlinear systems have shown that the proposed ENEM structure is better than the other results of RNN models.

Keywords: Elman network, Jordan network, dynamic system identification, multi-system identification, embedded memory

1. INTRODUCTION

Recurrent neural networks (RNNs) have been efficient identification tool in many areas since they have dynamic memories. A RNN with a dynamic memory is more suitable for representing a dynamic system, which has a dynamic mapping between its output(s) and input(s). RNNs generally require less neuron in the neural structure and less computation time. Moreover they have a low probability of being affected by external noise. Because of these features, RNNs have attracted the attention of researchers in the field of dynamic system identification [1-8].

RNNs can be classified in two categories: partially and fully recurrent. Two of the well-known partially RNNs are Elman and Jordan networks [9, 10]. These networks and their modified forms have been introduced to system identification [1, 2, 4, 6-8, 11-16]. The efficiency of the Elman network structure is limited to low order systems due to the insufficient memory capacity [1, 6, 17]. Elman network had failed in identifying even

Received August 23, 2004; revised December 8, 2004 & April 25 & June 7, 2005; accepted June 15, 2005.
Communicated by Pau-Choo Chung.

second order linear systems as reported in the references [1, 6, 9, 12, 16]. Several approaches have been suggested in the literature to increase the performance of the Elman network with simple modifications [1, 2, 14, 15, 17-21]. The suggested modifications on the Elman structure in the literature mostly have been able to improve certain kinds of problems. Although these modifications, it is not clear yet which network architecture is best suited to dynamic system identification.

In this paper, a new structural approach based on Elman network and NARX network is presented to improve the performance of Elman network on system identification. The new structure was called Elman network with embedded memory (ENEM). In order to show the performance of ENEM, two different identification approaches were considered. The first, a single ENEM identifier was used to identify single system. The second, a single ENEM identifier was used to identify multi-system having different orders with various numbers. The performance of ENEM was also compared to the results of Elman network, Jordan network and their modified forms.

The paper has five main sections. Section 2 briefly introduces partially RNNs, and the new ENEM structure. Section 3 discusses system identification, multi-system identification and adaptation of the new structure to multi-system identification. Section 4 shows experimental results. The work is finally concluded in section 5.

2. PARTIALLY RECURRENT NEURAL NETWORKS

2.1 Elman Network and Jordan Network

The RNN architectures range from fully interconnected to partially connected networks, including multilayer feedforward networks with distinct input and output layers. Fully connected networks do not have distinct input layers of nodes, and each node has input from all other nodes. Feedback to the node itself is possible. In the partially RNNs, although some nodes are part of a feedforward structure, other nodes provide the sequential context and receive feedback from other nodes. Weights from the context units are processed like those for the input units, for example, using backpropagation. The context units receive time-delayed feedback from the second layer units. Two fundamental ways can be used to add feedback into feedforward multilayer neural networks. Two of well known partially RNNs are Elman and Jordan networks [4].

Elman networks introduced feedback from the hidden layer to the context portion of the input layer. This approach pays more attention to the sequence of input values. Fig. 1 depicts the original Elman network with three layers of neurons [9]. $U(.)$, $X(.)$, $C(.)$, $Y(.)$, and W in the figure represent the inputs to the network, the outputs of the hidden units, the outputs of the context units, the outputs of the network, and weight vector respectively. The first layer of this network consists of two different groups of neurons. These are the group of external input neurons and the group of internal input neurons also called context units. Context units are also known as memory units as they store the previous output of the hidden neurons. In the Elman network, the values of the feedback connection weights have to be fixed by the user if the standard backpropagation (BP) algorithm is employed to train the network and usually their strengths are fixed at 1.0 [1, 6]. Theoretically an Elman network is able to model an nth order dynamic system [11].

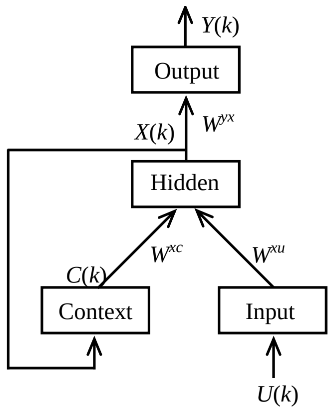


Fig. 1. Structure of the Elman network.

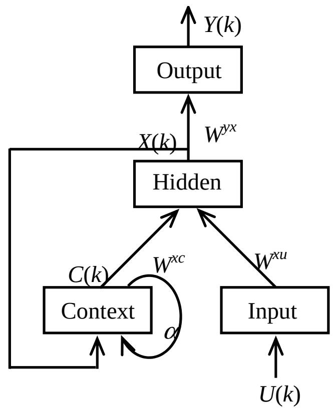


Fig. 2. A modified Elman network.

If self-connections are introduced to the context unit of a recurrent network, the ability of dynamic memorisation of the network can be increased [10, 22]. A modified Elman network proposed by Pham and Liu based on this idea is shown in Fig. 2. The values of the self-connection weights (α) are fixed between 0.0 and 1.0 before the training process. It is stated in [1] that the idea introducing self-feedback connections to the context unit was borrowed from the Jordan network.

Jordan RNNs use feedback from the output layer to the context nodes of the input layer and give more emphasis to the sequence of output values [10]. This neural network also has three layers and it has been theoretically shown that the original Jordan network is not capable of representing arbitrary dynamic systems [23]. However, by adding the feedback connections from the hidden layer to the context layer, similarly to the case of the Elman network, a modified Jordan network is obtained [1, 11]. These networks can be trained using the standard BP algorithm to model different dynamic systems. As with the modified Elman network, the values of the feedback connection weights have to be fixed by the user if the standard BP algorithm is employed.

2.2 Elman Network with Embedded Memory (ENEM)

The proposed new RNN is an extension of the Elman network. Pham and Liu reported that Elman network has memorization problem. To handle this problem, they suggested the new self-connection as shown in Fig. 2. The reason for this new connection was also to improve the dynamic memorization of the network [1].

There have been various network structures available in the literature to improve RNN performances. In these structures, network memorization capacity is limited and requires adjustments according to the problems applied. In reality, different systems might need different memorization capacity. It is thought that if a RNN having enough memorization for any system had been established, the identification would be sufficiently achieved. This might be acquired providing a dynamic memorization to each neural model according to the systems to be identified. In a way an established neural model might be even capable of identifying systems having long-term dependencies easily, effectively and accurately. This neural model also must be efficiently trainable with gradient-descent algorithms. To reach these objectives, dynamic memorization capabilities of the RNNs have to be increased.

In this study, a new RNN structure based on Elman network and NARX network memorization is presented. A new embedded memory (EM) unit has been added to the Elman structure and called as Elman network with embedded memory (ENEM). This new structure is illustrated in Fig. 3. The new EM unit consists of feedback connections from the network output to the input of the EM unit. The aim of adding this new EM unit to Elman network is to enlarge its dynamic memory capability. The output of this unit multiplied by W^{xd} is used as the input to the hidden unit. The number of inputs to this unit might be decided according to the system(s) to be identified.

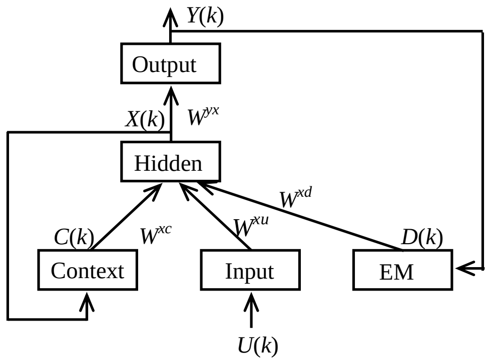


Fig. 3. Structure of ENEM.

The mathematical representation of the ENEM structure is summarized as

$$X(k) = F\{W^{xc}C(k),\ W^{xd}D(k),\ W^{xu}U(k)\} \tag{1}$$
$$C(k) = X(k-1) \tag{2}$$
$$D(k) = Y(k-1) \tag{3}$$
$$Y(k) = W^{xy}X(k) \tag{4}$$

where W^{xc}, W^{xu}, W^{xd} and W^{yx} are weight matrices and F is a function. $C(k)$ and $D(k)$ represent the context unit and the EM unit outputs, respectively.

As mentioned earlier, adding EM unit to Elman network should increase its dynamic memorisation. With the help of this unit, long-term dependencies of the Elman network should be improved. The idea of this EM unit has been borrowed from NARX network [24, 25]. The structure of this network with the memory of 3rd order is given in Fig. 4. A NARX network has the feedback connections from output unit to the input unit. The operation of the network is generally defined by

$$y(k) = F\{x(k), y(k-m), \ldots, y(k-1)\} \tag{5}$$

where m is the order of output memory, $y(k)$ is the output of the network, $x(k)$ is external inputs of the network, and F is a function [24, 25].

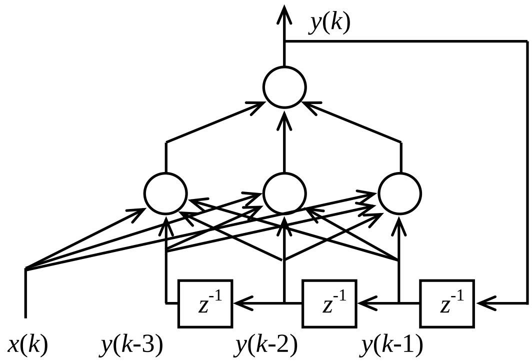


Fig. 4. A NARX network with output of 3rd order.

Lin *et al.* have shown that identifying the correct memory-order of a NARX network plays crucial roles in learning as well as generalization performance of a trained NARX network. They also reported that when the memory-order of a NARX network matches to the order of an unknown recursive system, the network improves learning and generalization performance [26, 27].

An EM unit in this work contains a NARX network memory structure having the same order with the systems to be identified. The structure of the mth order EM unit has been shown in Fig. 5 for the network having single output. The outputs of EM unit are fully connected to the hidden units of the Elman network. In this figure, $D(k)$ in Eq. (3) can be written as

$$D(k) = [y(k-1), y(k-2), \ldots, y(k-m)]^T. \tag{6}$$

This new EM unit presented in this work might enable more suitably enough memorization capacity and flexibility to the systems to be identified. It is assumed that this EM unit achieves mth order memorization.

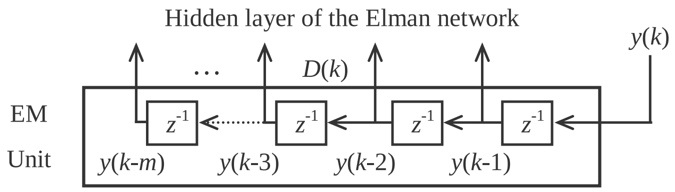


Fig. 5. Structure of m^{th} order EM unit.

3. DYNAMIC SYSTEM IDENTIFICATION

System identification is the process of constructing a model for an unknown system and estimating its parameters from experimental data. The behaviours of the systems can be linear or nonlinear. Linear systems are important in many respects. Most of the concepts and ideas for nonlinearity can be simplified in the form of linearity. Also, many control systems in industry are still being solved using linear techniques. An understanding of the details involved in the linear case allows the direct extension to solve nonlinear problems.

The input-output characteristics of systems generally change rapidly or even discontinuously when environment changes. If single identification model is used, the model requires self-adaptation to the new environment before an appropriate control action can be taken. In linear systems, such adaptation is possible, but the slowness of adaptation may result in a large transient error. Simpler model is always desired for both to identify the different environments as well as to control them rapidly and robustly.

3.1 Multi-System Identification

The multi-system identification scheme presented in this paper is also based on the same approach. Instead of identifying one system with only one identifier, a number of systems (multi-system) are identified using only one neural identifier in this work. Fig. 6 shows the neural identifier in diagrammatic form for multi-system identification. The input-output relationship of p number of dynamic systems is identified with single neural identifier as given in Fig. 6. The system identification approach is based on an adaptation with the tapped-delay line technique to increase its versatility and simplicity. The only knowledge required for the systems to be identified within one neural model is the number of systems.

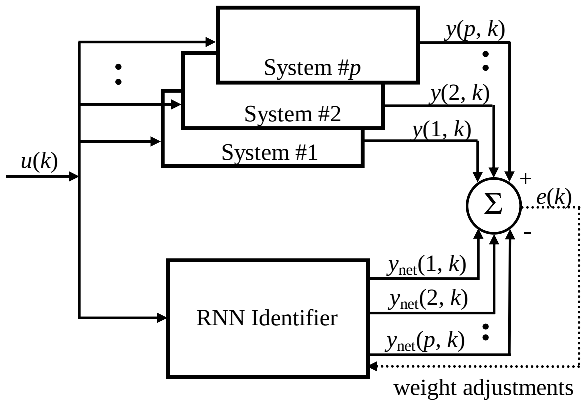


Fig. 6. RNN identifier for multi-system identification.

The mathematical form of multi-system is given as

$$Y(y_1(.), \ldots, y_p(.)) = A1y_1(.) + \ldots + APy_p(.) + Bu(.) \tag{8}$$

where $Y(.)$ depicts the outputs of multiple systems, $A1$, $A2$, …, AP and B represent the coefficients belonging to p number of different systems having same or different orders. As shown in Fig. 6, it is assumed that a single RNN might learn p number of dynamic systems. Identifying multi-system means achieving the coefficients giving in Eq. (8) with the help of single RNN identifier. For example, for multi-system identification, Eq. (8) becomes Eq. (9) when three of second order systems are considered as

$$Y(y_1(k), y_2(k), y_3(k)) = A1_1y_1(k-1) + A1_2y_1(k-2) + A2_1y_2(k-1) + A2_2y_2(k-2) + A3_1y_3(k-1) + A3_2y_3(k-2) + Bu(k-1). \quad (9)$$

For further identifications, the general formula given in Eq. (8) must be reorganised according to the number of systems to be identified or the system type to be identified. Throughout this work, RNN identifiers are trained in a supervised mode. The differences between the outputs of the neural identifier and the outputs of the systems are being used to modify the connection weights by the BP with momentum (BPM) algorithm.

For multi-system identification, EM unit configuration has to be adapted. In this configuration, the EM unit contains p number of NARX memory structures. These EM units might provide enough memorization capacity and flexibility to the multi-systems to be identified. The number of EM units is assigned according to the systems to be identified within single RNN.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In experimental studies, as stated earlier two different identification approaches were considered. Firstly, a single ENEM identifier was tested to identify only single system. Secondly, it was also tested to identify multi-system having the same or different orders with various numbers. The simulations results belonging to ENEM were compared to the results of Elman, modified Elman, Jordan, and modified Jordan networks. Total ten systems given in Table 1 were used throughout this study.

Table 1. Linear and nonlinear systems used for the proposed identification scheme.

System	System /Orders	Discrete-time domain representation of the systems
#1	Linear 1st order	$y(k+1) = 0.75 * y(k) + 0.25 * u(k)$
#2		$y(k+1) = 0.5 * y(k) + 0.1 * u(k)$
#3		$y(k+1) = 0.4 * y(k) + 0.3 * u(k)$
#4	Linear 2nd order	$y(k+1) = 1.752821 * y(k) - 0.818731 * y(k-1) + 0.011698 * u(k) + 0.010942 * u(k-1)$
#5		$y(k+1) = 1.1953 * y(k) - 0.4317 * y(k-1) + 0.1348 * u(k) + 0.1017 * u(k-1)$
#6		$y(k+1) = 1.8 * y(k) - 0.837 * y(k-1) + 0.019 * u(k) + 0.018 * u(k-1)$
#7	Linear 3rd order	$y(k+1) = 2.627771 * y(k) - 2.333261 * y(k-1) + 0.697676 * y(k-2) + 0.017203 * u(k) - 0.030862 * u(k-1) + 0.014086 * u(k-2)$
#8		$y(k+1) = 2.038 * y(k) - 1.366 * y(k-1) + 0.301 * y(k-2) + 0.0059 * u(k) - 0.018 * u(k-1) + 0.0033 * u(k-2)$
#9		$y(k+1) = 2.0549 * y(k) - 1.3524 * y(k-1) + 0.2894 * y(k-2) + 0.0049 * u(k) + 0.0032 * u(k-1)$
#10	Nonlinear 2nd order	$y(k) = 1.04 * y(k-1) - 0.824 * y(k-2) + 0.130667 * y^3(k-2) - 0.16 * u(k-2)$

In the simulations, the neural identifiers used in test were trained in a supervised mode with the BP with momentum (BPM) had one neuron in the input layer and six neurons in the hidden layers. Each system from system#1 to system#9 is represented with one neuron in the output layer and all neurons employed in the nine neural identifiers had linear activation functions. For the nonlinear system, the neural identifiers were configured with one neuron in the input layer and output layer and six neurons in the hidden layers. In the hidden layers tangent hyperbolic activation functions were selected.

The choice of the training input sequence is important for a successful RNN training. Sinusoidal and random input sequences are usually employed to get better performance in training [8]. In the current investigation, a random sequence of 200 input signals was adopted. The number of input signals was found to be a good compromise between a lower number which might be insufficient to represent the input-output behaviour of a plant and a larger number which lengthens the training process. A random signal in the range [− 1.0, 1.0] is used as the input to the systems and the networks in training and test processes. The training and test data files were obtained by applying uniform sequences of $u(k)$, $k = 1, 2, \dots, 200$. Each neural model was trained maximum 200,000 iterations or a specific time of 10 minutes. At the beginning of a training session, the weights of the neural connections were initialized to small random values in the range [− 0.2, 0.2] for the purpose of avoiding weight paralysis and speeding up the convergence of the networks.

4.1 Linear System Identification

The simulation results of the ENEM for the first, second and third order linear dynamic systems were given in Tables 2, 3 and 4, respectively. In each table, the same order systems which are evaluated as single, two or three together were considered. In Table 2, first order linear systems have been identified with single RNN. Systems #1, #2 and #3 have been identified with a RNN for single and multiple-system. In Tables 3 and 4, the same approaches have been followed for second and third order linear systems, respectively. During training, the learning coefficient (η) and the momentum coefficient (β) of the BPM were taken between 0.0005 and 0.05. These parameters were found after many trials (at least 20 trials) and given in Tables 2-4.

Table 2. Identification results for first order linear systems.

RNN	System	No of Sys.	η	β	RMS Error	Average RMS Error
#1	#1	One	0.002	0.015	0.005941	–
#2	#2	One	0.002	0.015	0.005428	–
#3	#3	One	0.002	0.015	0.007019	–
#4	#1 #2	Two	0.010	0.010	0.009501 0.003480	0.006491
#5	#1 #3	Two	0.002	0.015	0.013642 0.002420	0.008031
#6	#2 #3	Two	0.010	0.010	0.005911 0.014580	0.010245
#7	#1 #2 #3	Three	0.002	0.002	0.016476 0.004355 0.007387	0.009406

Table 3. Identification results for second order linear systems.

RNN	System	No of Sys.	η	β	RMS Error	Average RMS Error
#8	#4	One	0.0500	0.0500	0.056812	–
#9	#5	One	0.0100	0.0150	0.028961	–
#10	#6	One	0.0020	0.0020	0.032905	–
#11	#4 #5	Two	0.0020	0.0020	0.030097 0.035505	0.032801
#12	#4 #6	Two	0.0020	0.0020	0.007427 0.021997	0.014712
#13	#5 #6	Two	0.0020	0.0020	0.052802 0.042223	0.0475125
#14	#4 #5 #6	Three	0.0005	0.0005	0.029660 0.049646 0.072767	0.050691

Table 4. Identification results for third order linear systems.

RNN	System	No of Sys.	η	β	RMS Error	Average RMS Error
#15	#7	One	0.0200	0.0200	0.005476	
#16	#8	One	0.0200	0.0200	0.028389	
#17	#9	One	0.0100	0.0200	0.028919	
#18	#7 #8	Two	0.0020	0.0020	0.011400 0.046234	0.028817
#19	#7 #9	Two	0.0050	0.0050	0.007342 0.044646	0.025994
#20	#8 #9	Two	0.0050	0.0050	0.042211 0.064914	0.053562
#21	#7 #8 #9	Three	0.0005	0.0005	0.004876 0.021333 0.025175	0.017128

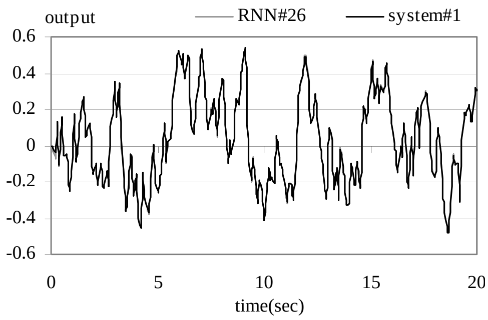
Simulations based on the second test focus on identifying three systems together having different orders. The identification results belonging to ENEM, and the other four networks for the systems (system#1, system#4, and system#7) were given in Table 5. The average RMS errors for three systems obtained from various RNN models and the improvement achieved from ENEM structure comparing to the other four networks have shown in Table 6. The responses of the systems and the ENEM identifiers (RNN#26) are illustrated in Fig. 7.

Table 5. Identification results for three systems together in one RNN model.

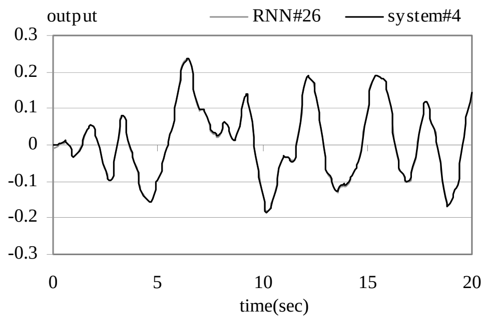
System No	RMS Errors				
	Elman (RNN#22)	Modified Elman (RNN#23)	Jordan (RNN#24)	Modified Jordan (RNN#25)	ENEM (RNN#26)
#1	0.006813	0.007721	0.044600	0.006795	0.003601
#4	0.042686	0.009629	0.029523	0.012338	0.001635
#7	0.004116	0.002191	0.004384	0.001555	0.000930

Table 6. Comparative results for RNN models.

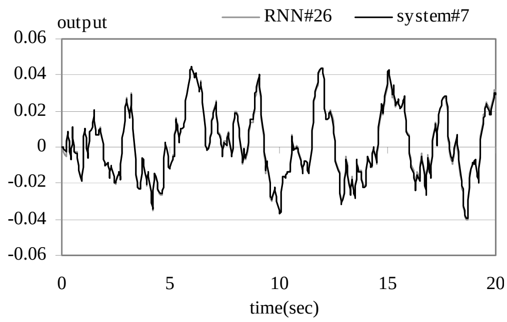
RNN models	Average RMS error	Improvement obtained from ENEM (%)	Parameters	
			η	β
Original Elman	0.017872	88.50	0.0020	0.0020
Modified Elman	0.006514	68.45	0.0001	0.0100
Original Jordan	0.026169	92.15	0.0010	0.0010
Modified Jordan	0.006896	70.20	0.0002	0.0002
ENEM	0.002055	–	0.0010	0.0010



(a) System #1 (first order, RMS error = 0.003601).



(b) System #4 (second order, RMS error = 0.001635).



(c) System #7 (third order, RMS error = 0.000930).

Fig. 7. Responses of ENEM identifier and different systems having different orders (average RMS error = 0.00205533).

Obviously, the results show that ENEM structure achieves the task better than the other four RNN models. The close identification results for the both individual and multi- system have shown that the proposed ENEM identifier is very successful and robust. As reported in [8-11], original Elman network had failed in identifying the second order linear systems. In contrast to the reports available in the literature, Elman network in this work was successfully trained not only for the second order systems but also third order systems and multiple systems. It needs to be emphasized that the selecting appropriate parameters become crucial point in Elman network training.

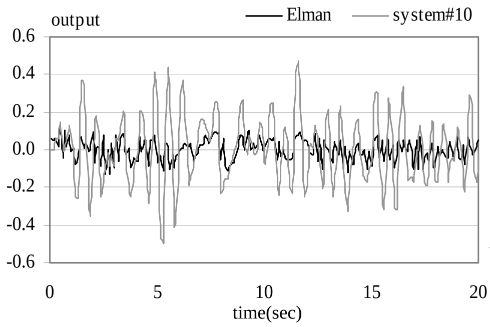
The RNN identifiers used for single and multiple systems having first order require maximum 100,000 iterations. The other RNN identifiers need 200,000 iterations for sufficient training.

4.2 Nonlinear System Identification

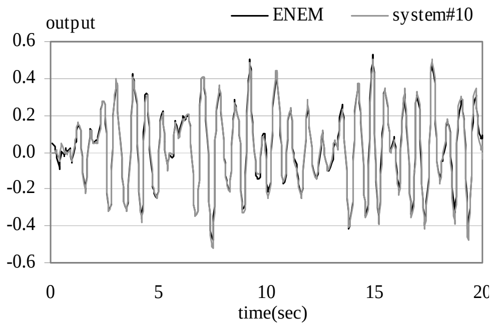
In order to examine the identification performance of the proposed network structure on a nonlinear system, a second-order nonlinear system (system#10) as indicated in Table 1 was also considered. This system is known as a simple pendulum swinging through small angles [11].

In order to show the robustness of the ENEM, ten different initial settings were applied to the neural identifiers for the system. The simulation results belonging to ENEM were compared to the result obtained from Elman network. The average RMS errors achieved from the identification process were 0.180213 and 0.031755 for Elman network and ENEM, respectively.

The identification results have shown that ENEM structure can demonstrate significant improvement on learning the nonlinear system behaviour as well. The adding EM unit to Elman network significantly increases identification performance about 85%. The responses achieved from Elman and ENEM networks for the nonlinear system were given in Fig. 8.



(a)



(b)

Fig. 8. (a) Responses of nonlinear system and Elman network (RMS error = 0.158873), (b) Responses of nonlinear system and ENEM (RMS error = 0.021520).

5. CONCLUSIONS

This work successfully presents a new partially RNN structure called ENEM for linear and nonlinear dynamic system identification. The identification results obtained from ENEM structure were also compared to the results of Elman network, Jordan network and their modified forms. The results have shown that ENEM is significantly better than the other four networks.

It is seen that Elman network can be trainable to identify for not only second order but also third order linear systems with acceptable error value by using suitable network parameters to the contrary published results in open literature. However, the adding EM unit to Elman network significantly increases identification performance for linear and nonlinear systems. It can be clearly stated that the EM units obviously increases learning and memorization capabilities of Elman network.

It can be finally concluded in general that ENEM structure is very suitable for both linear and nonlinear dynamic system identification. The ENEM structure could be easily and effectively adapted to other problems. So, it is expected that integrating EM unit into other partially RNNs might increase the performances of those networks for new applications.

REFERENCES

1. D. T. Pham and X. Liu, "Identification of linear and nonlinear dynamic systems using recurrent neural networks," *Artificial Intelligence in Engineering*, Vol. 8, 1993, pp. 90-97.
2. J. W. Dong, J. X. Qian, and Y. X. Sun, "Recurrent neural networks for recursive identification nonlinear dynamic process," in *Proceedings of the IEEE International Conference on Industrial Technology*, 1994, pp. 794-798.
3. C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Transactions on Neural Networks*, Vol. 6, 1995, pp. 144-156.
4. L. R. Medsker and L. C. Jain, *Recurrent Neural Networks: Design and Applications*, CRC Press, 2001.
5. C. A. L. Bailer-Jones, D. J. C. MacKay, and P. J. Withers, "A recurrent neural network for modelling dynamical systems," *Network: Computation in Neural Systems*, Vol. 9, 1998, pp. 531-547.
6. D. T. Pham and X. Liu, *Neural Networks for Identification*, *Prediction and Control*, 4th ed., Springer-Verlag, London, 1999.
7. D. T. Pham and D. Karaboga, "Training Elman and Jordan networks for system identification using genetic algorithms," *Artificial Intelligence in Engineering*, Vol. 13, 1999, pp. 107-117.
8. X. Ren and S. Fei, "Recurrent neural networks for identification of nonlinear systems," in *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 3, 2000, pp. 2861-2866.
9. J. L. Elman, "Finding structure in time," *Cognitive Science*, Vol. 14, 1990, pp. 179-211.
10. M. I. Jordan, "Attractor dynamics and parallelism in a connectionsist sequential machine," in *Proceedings of the 8th Conference on Cognitive Science*, 1986, pp. 531-546.
11. X. Liu, "Modelling and prediction using neural networks," Ph.D. Thesis, Department of System Engineering, University of Wales College of Cardiff, Cardiff, UK, 1993.
12. A. Kalinli and D. Karaboga, "Training recurrent neural networks by using parallel tabu search algorithm based on crossover operation," *Engineering Applications of Artificial Intelligence*, Vol. 17, 2004, pp. 529-542.
13. A. Kalinli and S. Sagiroglu, "A new approach based on recurrent neural networks for system identification," in *Proceedings of the International Symposium on Computer and Information Sciences*, LNCS 2869, Springer-Verlag, 2003, pp. 568-575.
14. D. P. Kwok, P. Wang, and K. Zhou, "Process identification using a modified Elman

neural net," *International Symposium on Speech, Image Processing and Neural Networks*, 1994, pp. 499-502.

15. X. Z. Gao, X. M. Gao, and S. J. Ovaska, "A modified Elman neural network model with application to dynamical systems identification," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, 1996, pp. 1376-1381.
16. D. T. Pham and X. Liu, "Training of Elman networks and dynamic system modelling," *International Journal of Systems Science*, Vol. 27, 1996, pp. 221-226.
17. W. Chagra, R. B. Abdennour, F. Bouani, M. Ksouri, and G. Favier, "A comparative study on the channel modeling using feedforward and recurrent neural network structures," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, 1998, pp. 3759-3763.
18. M. B. De Martino, "A partially recurrent architecture applied to classification problems," in *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 3, 1995, pp. 1244-1248.
19. T. B. Chen and V. W. Soo, "A comparative study of recurrent neural network architectures on learning temporal sequences," in *Proceedings of the IEEE International Conference on Neural Networks*, 1996, pp. 1945-1950.
20. P. Stagge and B. Sendhoff, "An extended Elman net for modeling time series," in *Proceedings of the International Conference on Artificial Neural Networks*, LNCS 1327, Springer-Verlag, 1997, pp. 427-432.
21. J. E. W. Holm and N. J. H. Kotze, "Training recurrent neural networks with leapfrog," in *Proceedings of the IEEE International Conference on Industrial Electronics*, 1998, pp. 99-104.
22. J. Hertz, A. Krogh, and R. G. Palmer, *Recurrent networks. In Introduction to the Theory of Neural Computation*, (Ch. 7), Addison-Wesley, Reading, 1991.
23. D. T. Pham and S. J. Oh, "A recurrent back propagation neural network for dynamic system identification," *Journal of Systems Engineering*, Vol. 2, 1992, pp. 213-223.
24. T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies is not as difficult with NARX recurrent neural networks," *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge, MA, 1996.
25. T. Lin, B. G. Horne, and C. L. Giles, "How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies," *Neural Networks*, Vol. 11, 1998, pp. 861-868.
26. T. Lin, B. G. Horne, C. L. Giles, and S. Y. Kung, "What to remember: how memory order affects the performance of NARX neural networks," in *Proceedings of the IEEE World Congress on Computational Intelligence*, Vol. 2, 1998, pp. 1051-1056.
27. C. L. Giles, T. Lin, B. G. Horne, and S. Y. Kung, "The past is important: a method for determining memory structure in NARX neural networks," in *Proceedings of the IEEE World Congress on Computational Intelligence*, Vol. 3, 1998, pp. 1834-1839.

Adem Kalinli received the B.S., M.S. and Ph.D. degrees in Electronics Engineering from Erciyes University, Turkey, in 1990, 1993 and 1996, respectively. He is now an Assistant Professor of Computer Division of Vocational High School at the Erciyes University of Turkey. He also holds various positions at Erciyes University. His current re-

search interests include intelligent optimisation techniques, neural networks and their engineering applications, particularly electronics engineering problems.

Seref Sagiroglu was born in August, 1965, in Antalya, Turkey. He received the B.S. degree in Electronic Engineering from Erciyes University, Kayseri, Turkey, in 1987. He received the Ph.D. degree in System Engineering from the University of Wales College of Cardiff, UK, in 1994. He worked as an associate professor between 1990 and 1993 at the Department of Computer Engineering, Erciyes University. He is now an Associate Professor at the Department of Computer Engineering, Gazi University. His research interests include modern heuristic optimization techniques (genetic algorithms, tabu searches, and simulated annealing), artificial neural networks, fuzzy logic, intelligent system identification, modelling and control, robotics, and computer and information security.